Bunte Bilder

u23 2012

andy, gordin

Chaos Computer Club Cologne e.V.
http://koeln.ccc.de

Cologne
2012-11-05

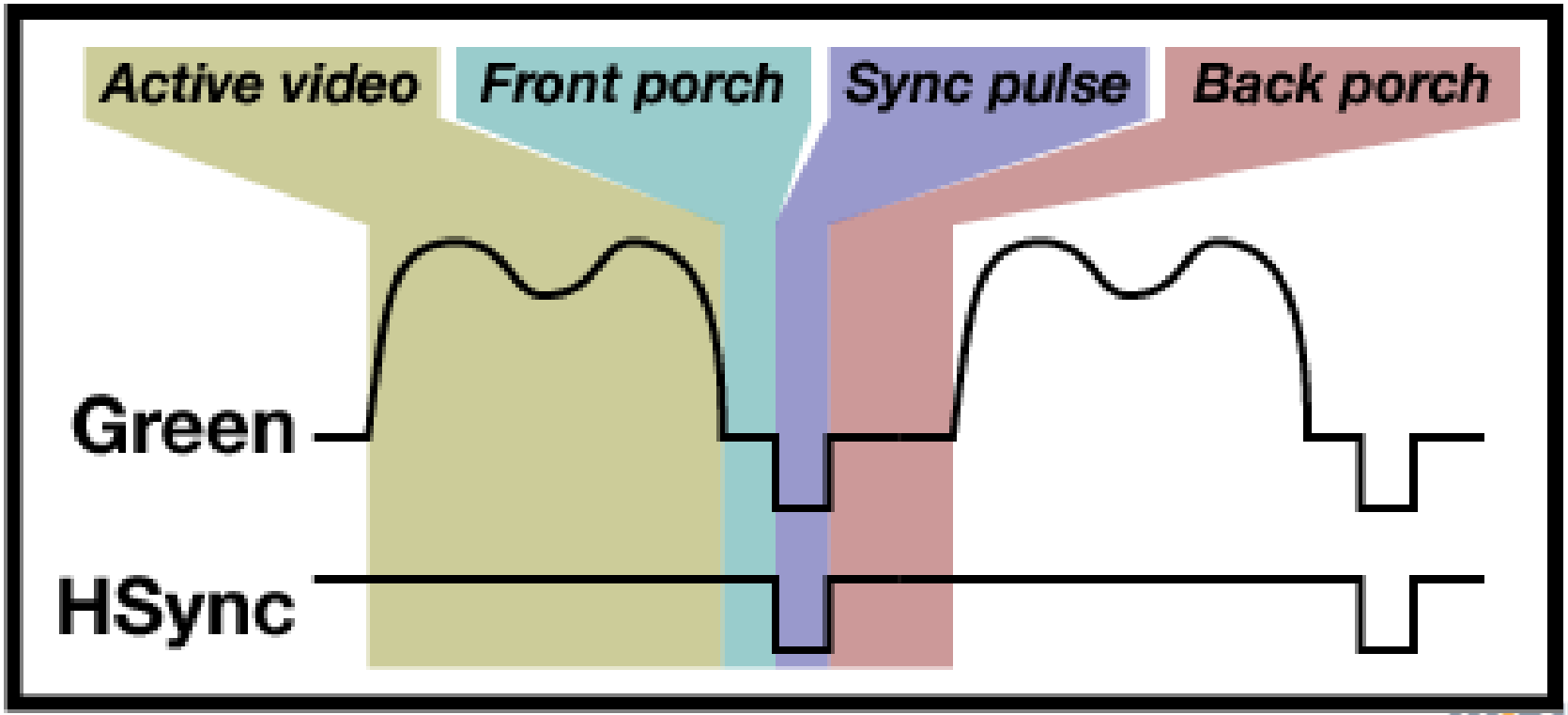
andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

VGA Signal

- (min.) 5 Signale:
  - Rot (analog max. 0.7V)
  - Grün (analog max. 0.7V)
  - Blau (analog max. 0.7V)
  - Horizontale Synchronisation
  - Vertikale Synchronisation
- optional 2 I2C-Leitungen für DDC (automatische Monitorerkennung)

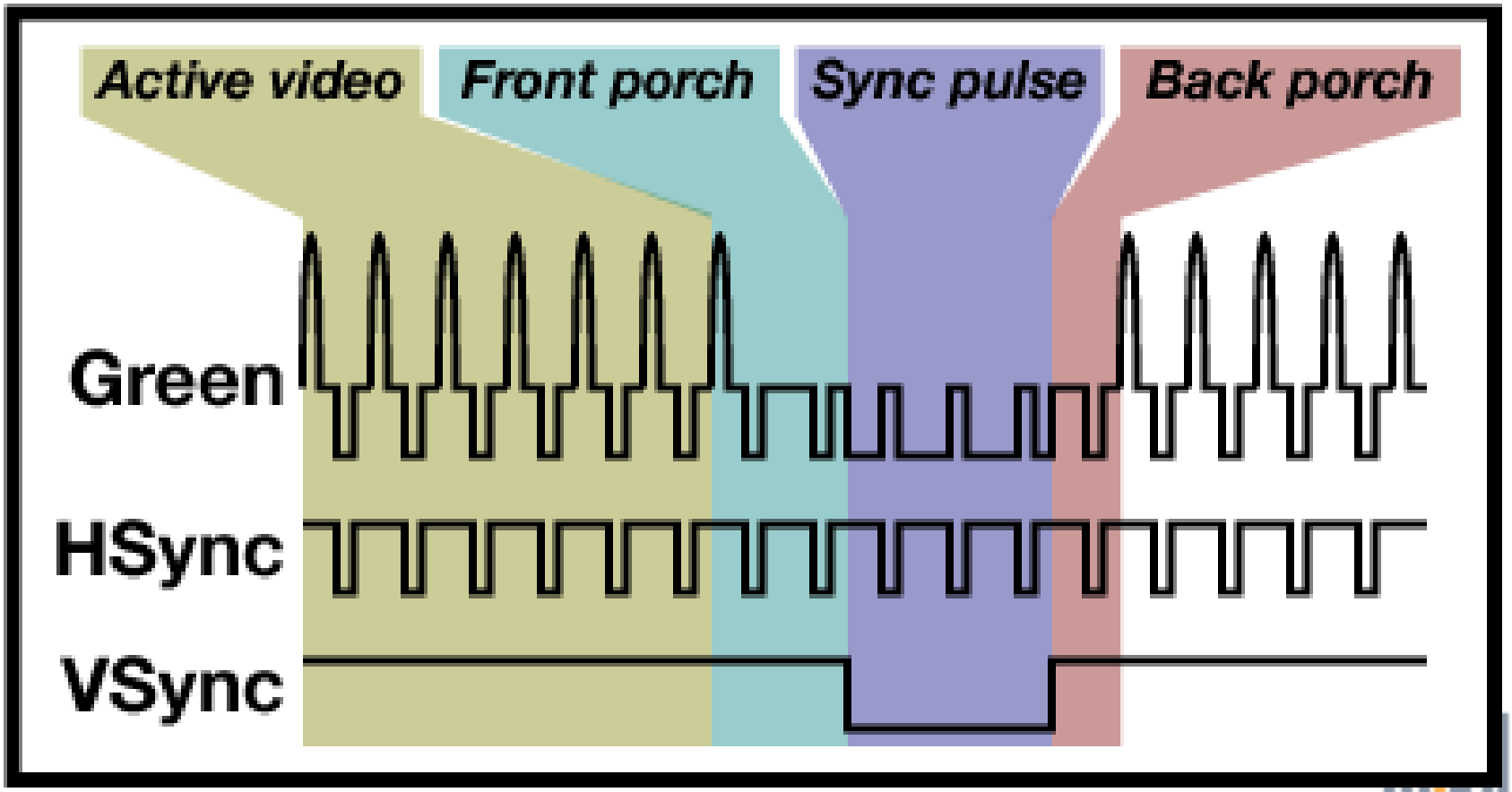
andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

VGA Signal



andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

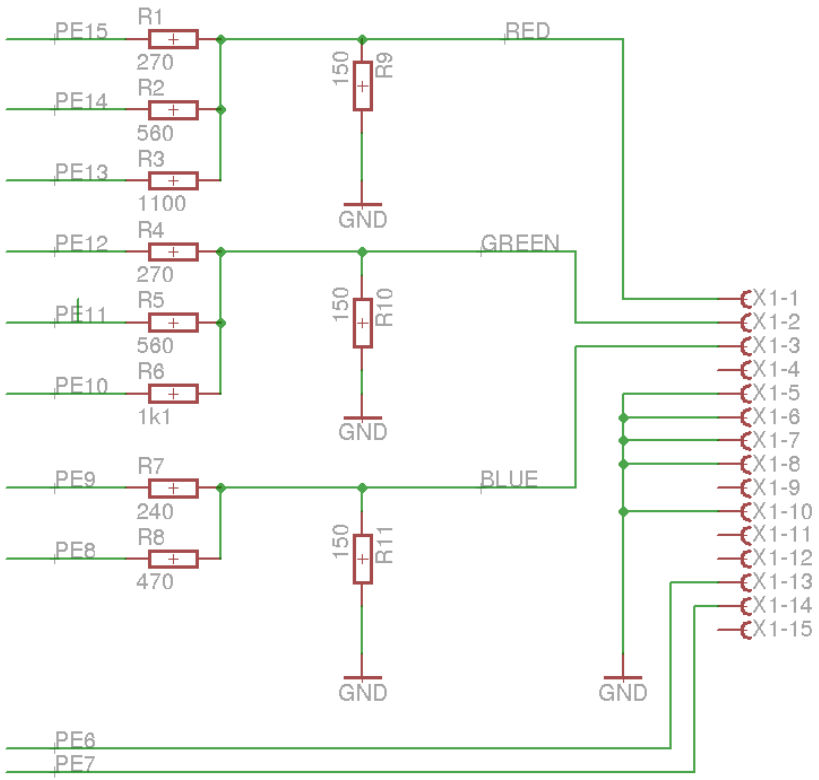
VGA Signal



andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

DAC

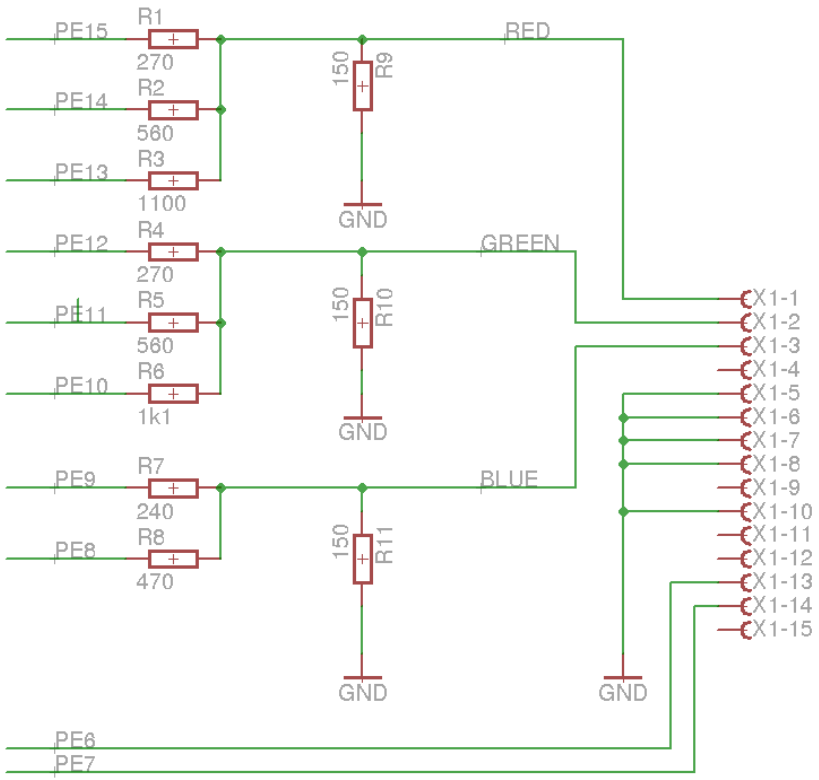
- **Problem:** Wir haben 8 Bit Farben in digitaler Form
- **Lösung:** Digital-Analog-Converter (DAC)
  - Auf unserer Platine ganz einfach mit 11 Widerständen:



andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

DAC

- **Problem:** Wir haben 8 Bit Farben in digitaler Form
- **Lösung:** Digital-Analog-Converter (DAC)
  - Auf unserer Platine ganz einfach mit 11 Widerständen:



andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Software Support

- VGA-Signal generiert durch libgaming (VGA.h, VGA.c)
- Pixeloperationen in libgraphics (Drawing.h, RLEBitmap.h):
  - Text ausgeben (Font.h)
  - Einzelne Pixel setzen/lesen
  - Horizontale/Vertikale/Diagonale Linien malen
  - Rechtecke (gefüllt und nicht gefüllt) malen
  - Kreise (gefüllt und nicht gefüllt) malen
  - Bitmaps (und Run-Length-Encodete Bitmaps) malen

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

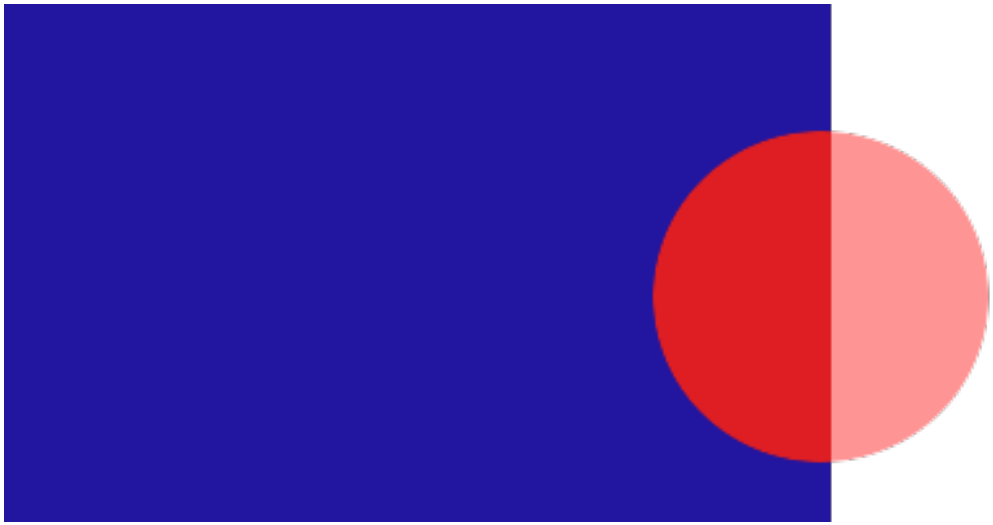
Software Support

- VGA-Signal generiert durch libgaming (VGA.h, VGA.c)
- Pixeloperationen in libgraphics (Drawing.h, RLEBitmap.h):
  - Text ausgeben (Font.h)
  - Einzelne Pixel setzen/lesen
  - Horizontale/Vertikale/Diagonale Linien malen
  - Rechtecke (gefüllt und nicht gefüllt) malen
  - Kreise (gefüllt und nicht gefüllt) malen
  - Bitmaps (und Run-Length-Encodete Bitmaps) malen

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Software Support

- Sonderformen der Draw-Funktionen:
  - NoClip: Bild wird nicht abgeschnitten, wenn es aus der Surface rausragen würde:



  - Composite: Funktion bekommt Callback in dem 2 Pixel übergeben werden, wo man entscheidet welche Farbe gemalt wird (nötig für Alphablending etc.):

```
typedef Pixel (*CompositionMode)(Pixel below, Pixel above);
```


andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Einfache Formen

```
void DrawFilledRectangle(Bitmap *bitmap, int x, int y, int w, int h, Pixel c);
```

- **Beispiel:** DrawFilledRectangle(surface, 10, 10, 50, 50, RGB(255,0,0));
- Malt ein gefülltes rotes Rechteck mit Ursprung bei (10,10) mit Größe 50x50 Pixeln

```
void DrawFilledCircle(Bitmap *bitmap, int x, int y, int r, Pixel c);
```

- **Beispiel:** DrawFilledCircle(surface, 10, 10, 25, RGB(255,0,0));
- Malt einen gefüllten roten Kreis mit Ursprung bei (10,10) mit Radius von 25 Pixeln

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Einfache Formen

```
void DrawFilledRectangle(Bitmap *bitmap, int x, int y, int w, int h, Pixel c);
```

- **Beispiel:** DrawFilledRectangle(surface, 10, 10, 50, 50, RGB(255,0,0));
- Malt ein gefülltes rotes Rechteck mit Ursprung bei (10,10) mit Größe 50x50 Pixeln

```
void DrawFilledCircle(Bitmap *bitmap, int x, int y, int r, Pixel c);
```

- **Beispiel:** DrawFilledCircle(surface, 10, 10, 25, RGB(255,0,0));
- Malt einen gefüllten roten Kreis mit Ursprung bei (10,10) mit Radius von 25 Pixeln

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Bitmaps malen

```
void DrawRLEBitmap(Bitmap *bitmap,const RLEBitmap *src,int x,int y);
```

- Für komplexere Bilder müssen Bilder konvertiert werden
- Converter in utils/ImageConverter
- Compilen mit *make*, dann:

```
./PNGConverter <dateiname>.png RLEBitmap <bildname> > <bildname>.c
```

- Oben in *bildname.c* einfügen:

```
#include <RLEBitmap.h>
```

- *bildname.c* in firmware Ordner kopieren und im Makefile eintragen
- in Files wo das Bitmap gemalt werden soll:

```
extern const RLEBitmap * const <bildname>;
```

- Bitmaps malen mit

```
DrawRLEBitmap(surface , <bildname>, 23, 42);
```

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Fonts erstellen (1)

- Tileset suchen oder selber malen
- http://dwarffortresswiki.org/index.php/Tileset_repository

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Fonts erstellen (2)

- Bild bearbeiten -> Hintergrund Transparent machen!
- PNG in Tools Ordner packen und konvertieren

```
./PNGConverter <font>.png RLEBitmapArray <fontname>
Fontbreite Fonthöhe spacing-x spacing-y
Bildbreite Bildhöhe > <fontname>.c
```

- Oben in *fontname.c* einfügen:

```
#include <game/Font.h>
#include <Bitmap.h>
#include <string.h>
```

- *fontname.c* in firmware Ordner kopieren
- *fontname.c* im firmware Makefile hinzufügen

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Font einstellen

```
void setFont(const RLEBitmap * const * font);
```

- Fonts sind monospace
- Fonts können unterschiedlich hoch und breit sein (wie sie im tileset sind)
- Standard Fonts: fontblack8, fontblack16, fontwhite8, fontwhite16
- für eigene Fonts hinzufügen wo man sie benutzen will:

  ```
  extern const RLEBitmap * const <fontname>[256];
  ```

- **Beispiel:**

  ```
  setFont(fontblack8);
  ```

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Text ausgeben

```
void DrawText(Bitmap* dest, char *text, int x, int y);
```

- Erst einen Font setzen (bleibt bis man einen anderen Font setzt)
- Nur ASCII (das was im tileset drin ist)
- Kann Zeilenumbrüche
- **Beispiel:** Schreibe foo an stelle 23, 42 und bar an Stelle 23, 42 + fontsize

```
DrawText(surface, "foo\nbar", 23, 42);
```

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Double Buffering

- Draw() Call im Spiel bekommt ein Bitmap übergeben (surface)
- In Bitmap kann reingemalt werden
- Nach jedem Frame wird dieses Bitmap gewechselt -> Double Buffering
- Während Frame 1 bemalt wird, wird Frame 2 auf VGA ausgegeben
- Danach andersrum
- Wichtig: Surface vor bemalen mit **ClearBitmap** clearen

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>

Hacken

Hackt Spiele!

andy <andy@koeln.ccc.de>, gordin <gordin@koeln.ccc.de>